



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/560,373	04/28/2000	Gregory Lucius Meredith	MS147248.1	3570

  

27195	7590	11/29/2007
AMIN. TUROCY & CALVIN, LLP		
24TH FLOOR, NATIONAL CITY CENTER		
1900 EAST NINTH STREET		
CLEVELAND, OH 44114		

  

EXAMINER	
KISS, ERIC B	

  

ART UNIT	PAPER NUMBER
2192	

  

NOTIFICATION DATE	DELIVERY MODE
11/29/2007	ELECTRONIC

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com  
hholmes@thepatentattorneys.com  
osteuball@thepatentattorneys.com

## Office Action Summary

Application No.

09/560,373

Applicant(s)

MEREDITH ET AL.

Examiner

Eric B. Kiss

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 07 September 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-22 and 28-33 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-22 and 28-33 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_

- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. The reply filed September 7, 2007, has been received and entered. Claims 1-22 and 28-33 are pending.

#### ***Response to Arguments***

2. Applicant's arguments filed September 7, 2007, have been fully considered but they are not persuasive.

Template discloses at least one of the child interdependent transactions is dependent on at least one other of the child interdependent transactions for completion (see e.g., *UsingWFT* at page 3-12; *DevelWFT* at pages 4-16 and 4-17 (completion of the order & requisition tasks requires the separate "Approve Requisition" and "Check Inventory" tasks to collectively commit, and thus, since the commitment of one task is insufficient to satisfy the AND junction criteria, the tasks are dependent upon each other for completion))

The "Install Solution" task, the overall parent task illustrated in Fig. 3-3 on page 3-12 of *UsingWFT*, is independent of the concurrent "Approve Requisition" and "Check Inventory" tasks; Note also that the "Approve Requisition" and "Check Inventory" are grouped together in a concurrent manner where they must collectively "commit" prior to the subsequent "Install Solution" task may begin, and thus may be considered child transactions as part of a parent transaction (see, e.g., *DevelWFT* on pages 4-16 and 4-17 describing the "AND" junction joining the ORDER and REQUISITION work items such that the destination task ("Install Solution") must wait until both associated work items are received)), and thus, the parent task commits separately from its child tasks.

Claim 1 does not require sending disparate work items to the destination tasks. Rather, claim 1 merely requires each child transaction receiving data that is at least partially different from data received by the other child transactions. Although the REQUISITION flow is identical for both tasks, the tasks receive other non-identical input (see, e.g., *DevelWFT*, page 5-11, describing other data used to create the ORDER work item (for example, the manager's name and the approval date))). In order for the manager's name and approval date to be affixed to the work item leaving the task, it must have been received as an input.

Template discloses a user interface allowing explicit definition of the above features. Specifically, all of the workflow elements of Template are defined in the Workflow Design Editor as described, for example, in pages 3-1 through 3-34 of *UsingWFT*.

***Claim Rejections - 35 USC § 103***

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.
4. Claims 1-14, 21, 22, and 28 are rejected under 35 U.S.C. 103(a) as obvious over Release 8.0 of the Workflow Template software product ("Template") publicly available from Template Software, Inc. in 1998 as evidenced by "Using the WFT Development Environment", 1998 (hereinafter *UsingWFT*) and "Developing a WFT Workflow System," 1998 (hereinafter *DevelWFT*) in view of "XML based Process Management Standard launched by Workflow Management Coalition – 'Wf-XML'," July 7, 1999 [online], accessed 01/03/2006, Workflow Management Coalition, <URL: <http://www.wfmc.org/pr/pr1999-07-07.pdf>>, 4 pages (hereinafter *WFXML-99*).

As per claim 1, Template is disclosed as reducing a business process using a programming language (workflow design; see *UsingWFT*, “Introduction” on page 3-2, and in particular, the first paragraph of that section);

dividing the reduced business process into at least one independent transaction and at least one parent transaction, the at least one independent transaction is not interdependent with the at least one parent transaction, the at least one parent transaction has two or more child interdependent transactions that are each different from each other and interdependent with each other, each child transaction receiving data that is at least partially different from data received by the other child transactions, the child transactions are children of the parent transaction (see *UsingWFT*, “Creating copy flows” on page 3-20 for distinguishing between concurrent autonomous (using separate flows) business operations and concurrent interdependent (using a single flow) business operations (the copy flow allows operations using the same flow to be represented independently; see, for example, *UsingWFT*, Fig. 3-3 on page 3-12 in which the copy flow junction box supplies the same “REQUISITION” flow to both the “Approve Requisition” and “Check Inventory” tasks; see also *UsingWFT*, “Creating compound flows” on page 3-19 for grouping business operations into concurrent interdependent transactions (forms a work item set associated with the compound flow); note that the “Approve Requisition” and “Check Inventory” tasks are non-uniform, disparate, pluralistic tasks, despite the “REQUISITION” flow provided to each task being identical; Note that the “Install Solution” task, also illustrated in Fig. 3-3 on page 3-12 of *UsingWFT*, is independent of the concurrent “Approve Requisition” and “Check Inventory” tasks; Note also that the “Approve Requisition” and “Check Inventory” are grouped together in a concurrent manner where they must

Art Unit: 2192

collectively “commit” prior to the subsequent “Install Solution” task may begin, and thus may be considered child transactions as part of a parent transaction (see, e.g., *DevelWFT* on pages 4-16 and 4-17 describing the “AND” junction joining the ORDER and REQUISITION work items such that the destination task (“Install Solution”) must wait until both associated work items are received); further, although the REQUISITION flow is identical for both tasks, the tasks receive other non-identical input (see, e.g., *DevelWFT*, page 5-11, describing other data used to create the ORDER work item (for example, the manager’s name and the approval date)); wherein at least one of the child interdependent transactions is dependent on at least one other of the child interdependent transactions for completion (see e.g., *UsingWFT* at page 3-12; *DevelWFT* at pages 4-16 and 4-17 (completion of the order & requisition tasks requires the separate “Approve Requisition” and “Check Inventory” tasks to collectively commit, and thus, since the commitment of one task is insufficient to satisfy the AND junction criteria, the tasks are dependent upon each other for completion));

executing the at least one independent transaction independently from the at least one parent interdependent transaction to increase throughput and decrease latency of the business process, the at least one independent transaction committing after all child interdependent transactions have committed (forming a concatenation of the two or more input work items, as a result of an *And* junction condition; see, for example, *UsingWFT*, “Creating compound flows” on page 3-19; see also *DevelWFT* on pages 4-16 and 4-17 describing the “AND” junction joining the ORDER and REQUISITION work items such that the destination task (“Install Solution”) must wait until both associated work items are received (*i.e.*, the “Check Inventory” and “Approve Funds” transactions have committed)); and

transferring committed data associated with the at least one independent transaction and the at least one parent interdependent transaction to a computer component for further processing (see, for example, *UsingWFT*, “Creating compound flows” on page 3-19).

Template further discloses a user interface allowing explicit definition of the above features. Specifically, all of the workflow elements of Template are defined in the Workflow Design Editor as described, for example, in pages 3-1 through 3-34 of *UsingWFT*.

Template is not explicitly disclosed as the programming language having an XML syntax. However, *WFXML-99* teaches that workflow specifications may be written in such a programmable language having an XML syntax (Wf-XML; see, for example, the figure on p. 2 and the last paragraph of p. 2, continuing onto p. 3). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to include a programmable language having an XML syntax as once taught by *WFXML-99*. One would be motivated to do so to provide a robust tool for specifying workflows.

As per claims 2-3, Template is further disclosed as the children interdependent transactions respectively including one or more actions, the one or more actions being concurrently executed independently from each other, the respective children independent transactions committing when all of their associated actions are completed (see, for example, *UsingWFT*, Table 3-1 on page 3-3 and second paragraph of “About the Task Editor perspective on tasks” on page 6-2; and *UsingWFT*, “Creating compound flows” on page 3-19).

As per claim 4, Template is further disclosed as explicitly defining transaction boundaries for the at least one independent transaction and the children interdependent transactions as a function of a number of actions within the at least one independent transaction and the children

Art Unit: 2192

interdependent transactions, respectively, in order to define a granularity at an action level (a flow defines a possible route between tasks through which a work item can travel; see *UsingWFT*, Table 3-1 on page 3-3).

As per claim 5, Template is further disclosed as the children interdependent transactions being concurrently executed in isolation from each other (see, for example, *UsingWFT*, Table 3-1 on page 3-3 and “Creating copy flows” on page 3-20).

As per claim 6, Template is further disclosed as employing separate machines to execute the at least one independent transaction and the at least one parent interdependent transaction (see, for example, *UsingWFT*, Table 3-1 on page 3-3 and “Creating copy flows” on page 3-20).

As per claim 7, Template is disclosed as a user interface component (Workflow Design Editor) and a plurality of model components (tasks, flows, work items, roles, junctions, and labels) accessible through the user interface component and adapted to allow a user to create a model of a business process (workflow design; see *UsingWFT*, “Introduction” on page 3-2, and in particular, the first paragraph of that section), the plurality of model components comprising a distinguishing model component (copy flow junction box; see *UsingWFT*, “Creating copy flows” on page 3-20) for distinguishing between autonomous (using separate flows) business operations and interdependent (using a single flow) business operations, the autonomous business operations are not dependent on each other for completion and are concurrent with respect to each other, the interdependent business operations are dependent on each other for completion and are concurrent with respect to each other, the interdependent business operations being non-identical and each receiving data that is at least partially different from each other (the copy flow allows operations using the same flow to be represented independently; see *UsingWFT*, Fig. 3-3



Art Unit: 2192

on page 3-12 in which the copy flow junction box supplies the same “REQUISITION” flow to both the “Approve Requisition” and “Check Inventory” tasks; note that the “Approve Requisition” and “Check Inventory” tasks are non-uniform, disparate, pluralistic tasks, despite the “REQUISITION” flow provided to each task being identical; Note that the “Install Solution” task, also illustrated in Fig. 3-3 on page 3-12 of *UsingWFT*, is independent of the concurrent “Approve Requisition” and “Check Inventory” tasks; Note also that the “Approve Requisition” and “Check Inventory” are grouped together in a concurrent manner where they must collectively “commit” prior to the subsequent “Install Solution” task may begin, and thus may be considered child transactions as part of a parent transaction (see, e.g., *DevelWFT* on pages 4-16 and 4-17 describing the “AND” junction joining the ORDER and REQUISITION work items such that the destination task (“Install Solution”) must wait until both associated work items are received); further, although the REQUISITION flow is identical for both tasks, the tasks receive other non-identical input (see, e.g., *DevelWFT*, page 5-11, describing other data used to create the ORDER work item (for example, the manager’s name and the approval date))). Template is not explicitly disclosed as the software comprising a programmable language having an XML syntax. However, *WFXML-99* teaches that workflow specifications may be written in such a programmable language having an XML syntax (Wf-XML; see, for example, the figure on p. 2 and the last paragraph of p. 2, continuing onto p. 3). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to include a programmable language having an XML syntax as once taught by *WFXML-99*. One would be motivated to do so to provide a robust tool for specifying workflows.

As per claim 8, Template is further disclosed as a transaction grouping model component (compound flow junction box) for grouping business operations into interdependent business operations (forms a work item set associated with the compound flow; see *UsingWFT*, “Creating compound flows” on page 3-19).

As per claim 9, Template is further disclosed as the grouping model component (compound flow junction box) providing synchronization of interdependent business operations based on the completion of the interdependent business operations (forming a concatenation of the two or more input work items, as a result of an *And* junction condition; see *UsingWFT*, “Creating compound flows” on page 3-19).

As per claims 10 and 11, Template is further disclosed as associating actions (tasks) with transactions (work items; see *UsingWFT*, Table 3-1 on page 3-3 and second paragraph of “About the Task Editor perspective on tasks” on page 6-2). Therefore, the transaction grouping model component disclosed by *UsingWFT* also functions as an action grouping model as claimed.

As per claim 12, Template is further disclosed as the plurality of model components comprising at least one boundary establishing component (flows) for defining transaction (work item) boundaries (a flow defines a possible route between tasks through which a work item can travel; see *UsingWFT*, Table 3-1 on page 3-3).

As per claim 13, Template is further disclosed as a component for establishing concurrent operations (copy flow; see *UsingWFT*, Table 3-1 on page 3-3 and “Creating copy flows” on page 3-20).

As per claim 14, Template is further disclosed as a component for establishing sequential operations (plain flow; see *UsingWFT*, Table 3-1 on page 3-3).

As per claim 21, as admitted prior art, it was well known and commonly practiced in the computer art at the time the invention was made to incorporate a computer readable medium into a computer system in order to allow data transfer between the medium and the system, such as, for example, for the execution of a program embodied in a CD-ROM medium on such a computer system. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to have a computer readable medium residing on a computer system as part of a system incorporating the Template product.

As per claim 22, Template is further disclosed as the plurality of model components comprising a component (compound flow junction box) for defining concurrent synchronizing constraints as occurring upon the completion of the autonomous operations (forming a concatenation of the two or more input work items, as a result of an *And* junction condition; see *UsingWFT*, "Creating compound flows" on page 3-19).

As per claim 28, Template is disclosed as means for: distinguishing between synchronization of autonomous operations (using separate flows) and interdependent operations (using a single flow), the autonomous operations are not dependent on each other for completion and are concurrent with respect to each other, the interdependent operations are dependent on each other for completion and are concurrent with respect to each other, the interdependent operations each receive data that is at least partially dissimilar with respect to data received by each interdependent operation (the copy flow allows operations using the same flow to be represented independently; see *UsingWFT*, Fig. 3-3 on page 3-12 in which the copy flow junction box supplies the same "REQUISITION" flow to both the "Approve Requisition" and "Check Inventory" tasks; note that the "Approve Requisition" and "Check Inventory" tasks are

Art Unit: 2192

non-uniform, disparate, pluralistic tasks, despite the “REQUISITION” flow provided to each task being identical); expressing synchronization constraints on completion of autonomous concurrent operations (forming a concatenation of the two or more input work items, as a result of an *And* junction condition; see *UsingWFT*, “Creating compound flows” on page 3-19); and associating transaction operations and groups of business operations (creating a workflow design that represents the flow of work throughout your business; see *UsingWFT*, “Introduction” on page 2-2; Note that the “Install Solution” task, also illustrated in Fig. 3-3 on page 3-12 of *UsingWFT*, is independent of the concurrent “Approve Requisition” and “Check Inventory” tasks; Note also that the “Approve Requisition” and “Check Inventory” are grouped together in a concurrent manner where they must collectively “commit” prior to the subsequent “Install Solution” task may begin, and thus may be considered child transactions as part of a parent transaction (see, e.g., *DevelWFT* on pages 4-16 and 4-17 describing the “AND” junction joining the ORDER and REQUISITION work items such that the destination task (“Install Solution”) must wait until both associated work items are received); further, although the REQUISITION flow is identical for both tasks, the tasks receive other non-identical input (see, e.g., *DevelWFT*, page 5-11, describing other data used to create the ORDER work item (for example, the manager’s name and the approval date)). Template is not explicitly disclosed as the software comprising a programmable language having an XML syntax. However, *WFXML-99* teaches that workflow specifications may be written in such a programmable language having an XML syntax (Wf-XML; see, for example, the figure on p. 2 and the last paragraph of p. 2, continuing onto p. 3). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to include a

Art Unit: 2192

programmable language having an XML syntax as once taught by *WFXML-99*. One would be motivated to do so to provide a robust tool for specifying workflows.

5. Claims 15-20 and 29-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Template (as evidenced by *UsingWFT* and *DevelWFT*) in view of *WFXML-99*, as applied to claims 1 and 12 above, and further in view of U.S. Patent No. 5,940,839 to Chen et al.

As per claim 15, Template is disclosed as such a system for business process modeling including a user interface and a plurality of model components (see disclosure applied above to claim 12) but fails to teach a compensation model component adapted to compensate committed interdependent concurrent transactions and being invoked upon the occurrence of a failed interdependent concurrent transaction. However, Chen teaches, as part of a transaction processing method and system, such a compensation model component (transaction management system (TMS) mechanisms; see column 5, lines 10-48) adapted to compensate committed interdependent concurrent transactions and being invoked upon the occurrence of a failed interdependent concurrent transaction (see column 2, line 65 through column 3, line 33). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to incorporate a compensation model component as once taught by Chen. One would be motivated to do so to provide the ability to handle transaction failures.

As per claim 16, Chen further teaches transactions being children in a parent transaction (as part of an “ancestor tree”; see column 3, lines 24-27) wherein a compensation routine is invoked by the parent transaction (the failed transaction is undone by proceeding from the in-

Art Unit: 2192

process closest recoverable ancestor (ICRA) transaction; see column 3, lines 11-33). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to further modify the Template product to include invocation of a compensation model component by a parent transaction as per the teachings of Chen. One would be motivated to do so allow recovery of a failed transaction by reverting back to a parent transaction.

As per claim 17, Chen further teaches calling compensation routines within the committed interdependent concurrent transactions (see column 9, lines 4-17). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to further modify the Template product to include compensation routines within committed interdependent transactions as per the teachings of Chen. One would be motivated to do so enable elimination of the effect of a transaction.

As per claims 18-20, Chen further teaches calling compensation routines within a failed transaction based on information on committed transactions stored within a database (see column 8, line 61 through column 9, line 5). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to further modify the Template product to include the compensation model component calling compensation routines within the failed interdependent concurrent transaction based on information on the committed interdependent concurrent transactions stored within a database as per the teachings of Chen. One would be motivated to do so allow for compensation of committed transactions beyond the failure affected scope.

As per claims 29 and 30, Template is disclosed as such a method for business process modeling but fails to expressly disclose failing the at least one parent interdependent transaction when at least one of its children interdependent transactions does not commit, and compensating the at least one failed child transaction, the at least one parent interdependent transaction invoking a compensation routine within the at least one failed child transaction that compensates the at least one failed child transaction; failing the at least one parent interdependent transaction when at least one of its children interdependent transactions does not commit, and compensating the at least one failed child transaction, the at least one parent interdependent transaction invoking a compensation routine within the at least one failed child transaction that compensates the at least one failed child transaction. However, Chen teaches, as part of a transaction processing method and system, such a compensation model component (transaction management system (TMS) mechanisms; see, for example, column 5, lines 10-48) adapted to compensate committed interdependent concurrent transactions and being invoked upon the occurrence of a failed interdependent concurrent transaction (see, for example, column 2, line 65 through column 3, line 33). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to incorporate a compensation model component as once taught by Chen. One would be motivated to do so to provide the ability to handle transaction failures. Chen further teaches calling compensation routines within a failed transaction based on information on committed transactions stored within a database (see, for example, column 8, line 61 through column 9, line 5). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to further modify the Template product to include the compensation model component

Art Unit: 2192

calling compensation routines within the failed interdependent concurrent transaction based on information on the committed interdependent concurrent transactions as per the teachings of Chen. One would be motivated to do so allow for compensation of committed transactions beyond the failure affected scope.

As per claim 31, Template is disclosed as such a method for business process modeling but fails to expressly disclose compensating the at least one parent independent transaction when it does not commit and all of its children interdependent transactions commit. However, Chen teaches, as part of a transaction processing method and system, such a compensation model component (transaction management system (TMS) mechanisms; see, for example, column 5, lines 10-48) adapted to compensate a parent uncommitted independent transactions and being invoked upon the occurrence of a failed interdependent child transaction (see, for example, column 2, line 65 through column 3, line 33; and col. 8, line 60, through col. 9, line 26). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to incorporate such a compensation model component as once taught by Chen. One would be motivated to do so to provide the ability to handle transaction failures and to allow for compensation of transactions.

As per claims 32 and 33, Template is disclosed as such a method for business process modeling but fails to expressly disclose compensating the at least one parent interdependent transaction when it does not commit and all of its children interdependent transactions commit, the at least one parent interdependent transaction invoking its own compensation routine. However, Chen teaches, as part of a transaction processing method and system, such a compensation model component (transaction management system (TMS) mechanisms; see, for



Art Unit: 2192

example, column 5, lines 10-48) adapted to compensate a parent uncommitted interdependent transactions and being invoked upon the occurrence of a failed interdependent child transaction (see, for example, column 2, line 65 through column 3, line 33; and col. 8, line 60, through col. 9, line 26). Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to modify the Template product to incorporate such a compensation model component as once taught by Chen. One would be motivated to do so to provide the ability to handle transaction failures and to allow for compensation of transactions.

### *Conclusion*

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

7. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

Art Unit: 2192

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist: 571-272-2100.



Eric B. Kiss  
Primary Patent Examiner  
November 23, 2007